Informatique industrielle

CodeWarrior HC12 + HIWAVE
Boutons + **LED** + LCD

TP ii 1

OBJECTIFS : Maîtriser l'environnement de développement croisé Metrowerks :

Créer un projet CodeWarrior	Corriger les erreurs	Utiliser l'écran LCD,
Editer les sources,	Charger le programme,	Gérer les LED
Compiler les sources	Utiliser le débogueur,	Tester les boutons.

PREREQUIS :

Identifier les éléments du langage C.	Lire et écrire avec un pointeur à une adresse.
Appeler une fonction avec des paramètres.	Manipuler les bits avec des masques.
Lire et écrire sur un port I/O avec des macros prédéfinies.	

NOTES :
Ce document comprend trois parties. Un tutorial complètement guidé, une série d'exercices simples, un guide de dépannage.

AUTEUR :

Jacques-Olivier KLEIN

RETOUR d'EXPERIENCE :

Aucun pour l'instant.

DATE de REVISION :
Version initiale : 20 juillet 2004
Révision : 24 août 04

A Tutorial CodeWarrior

A.1 Lancer l'application CodeWarrior pour HC12

Menu Démarrer->Programmes->Metrowerks->Codewarrior MOT V1.2-> CodeWarrior IDE

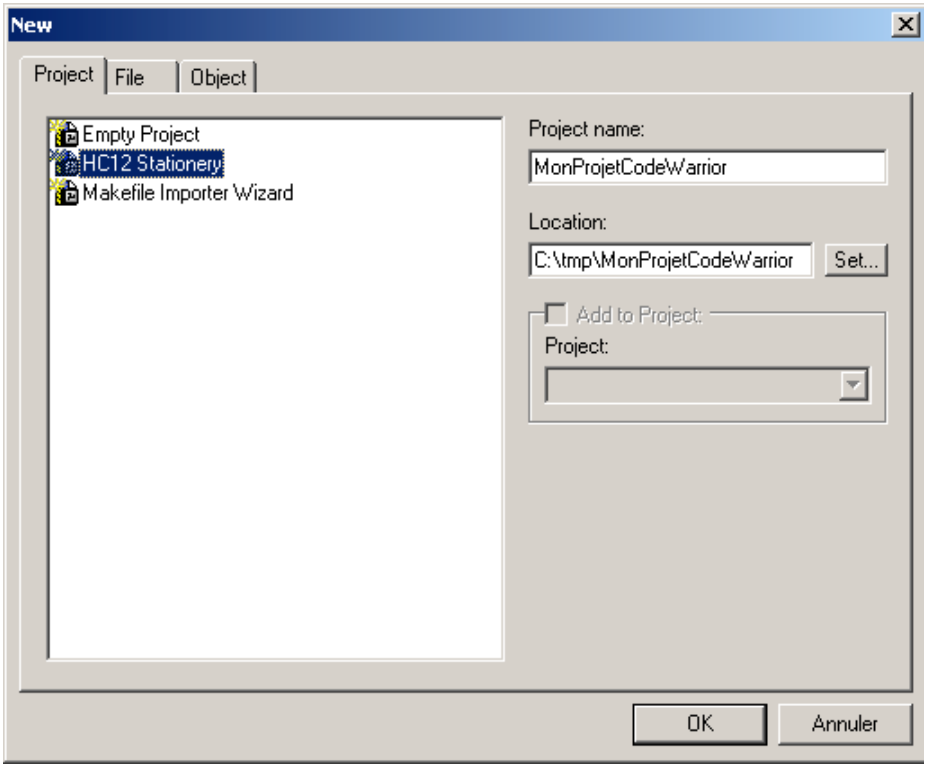
ou dans la barre d'outils .

La barre de menu principale de CodeWarrior apparaît. Vous ne fermerez cette fenêtre que pour quitter CodeWarrior.

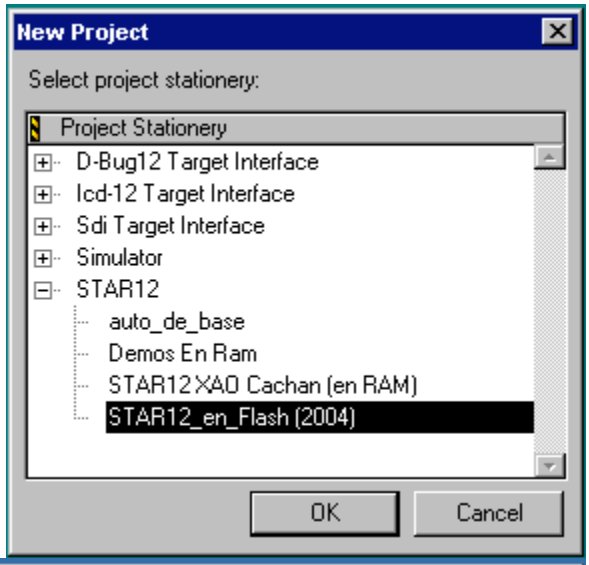
A.2 Créer un nouveau projet à partir d'un projet de base

File->New... ou La fenêtre *New* s'ouvre.

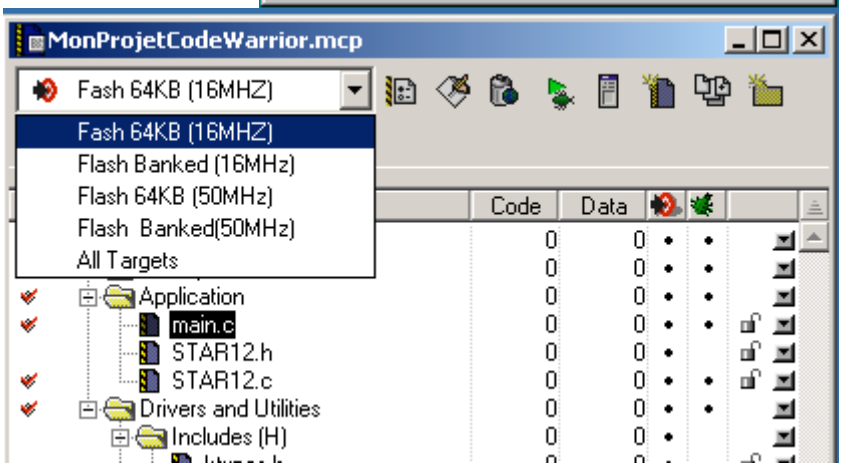
Dans la fenêtre *New*, sélectionner l'onglet *project*, choisir *HC12 Stationnery. A*vec le bouton *[Set...], c*hoisir un répertoire pour enregistrer le projet, de préférence sur le disque local, c:\tmp est un bon choix. Nommer votre projet dans le champs *project name*. Vérifier le type de projet (HC12 stationnery), le chemin (C:\tmp\MonProjet) et le nom de votre projet puis valider par le bouton *[OK]*.



La fenêtre *New Project* s'ouvre. Sélectionner un stationnery correspondant à la carte (STAR12) utilisée, par exemple *STAR12-> STAR12 en Flash* puis valider par le bouton *[OK]*.



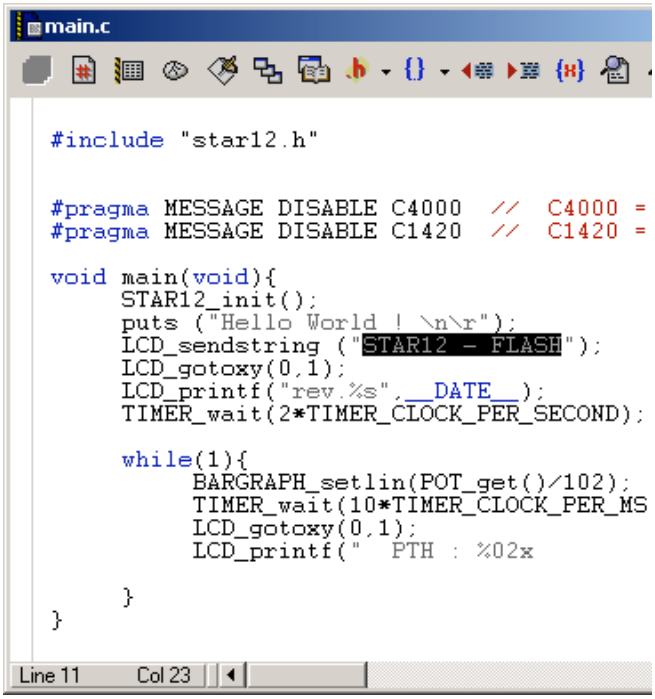
Une nouvelle fenêtre de projet s'ouvre. Vous pouvez choisir différentes configurations appelées "cibles".
- Flash signifie que le programme ira en mémoire flash.
- 16 MHz ou 50 MHz doit correspondre à la fréquence de l'oscillateur installé sur votre carte STAR12.
- Enfin préférez 64KB à Banked dans un premier temps. 64KB correspond à un espace mémoire linéaire de 64KB alors que Banked correspond à un espace mémoire paginé.



A.3 Éditer un fichier source

Dans la fenêtre de projet (voir feuille précédente), sélectionner l'onglet *Files*, développer les groupes (en forme de dossier) et y repérer le fichier main.c.
L'ouvrir en double cliquant dessus.
Expérimentez les différents outils de l'éditeur de texte :

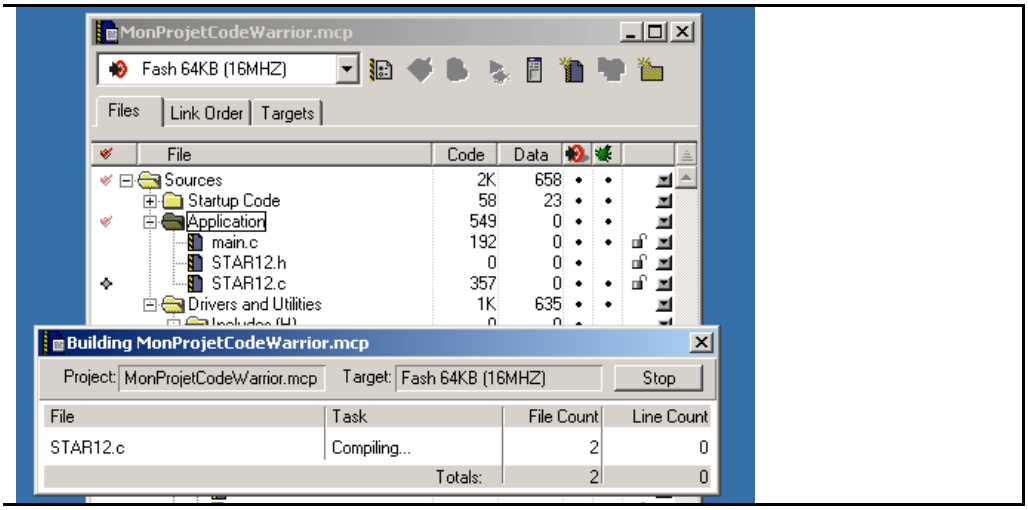
et n'oubliez pas que vous pouvez toujours annuler plusieurs opérations grâce à la touche [CTRL-Z].
Après une première compilation, vous verrez de la couleur correspondant aux différents éléments syntaxiques, par exemple, les fonctions en rouge vif, les commentaires en rouge sombre, les mot-clé du langage en bleu , les macros en vert, les types et les variables en bleu clair.



A.4 Générer l'exécutable

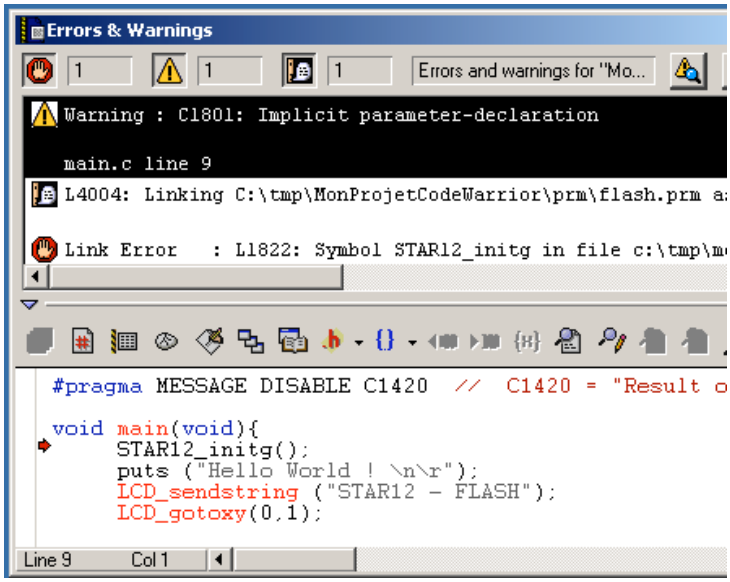
Project -> Make

ou [F7] ou .
Une fenêtre présente l'état d'avancement de la compilation du projet.



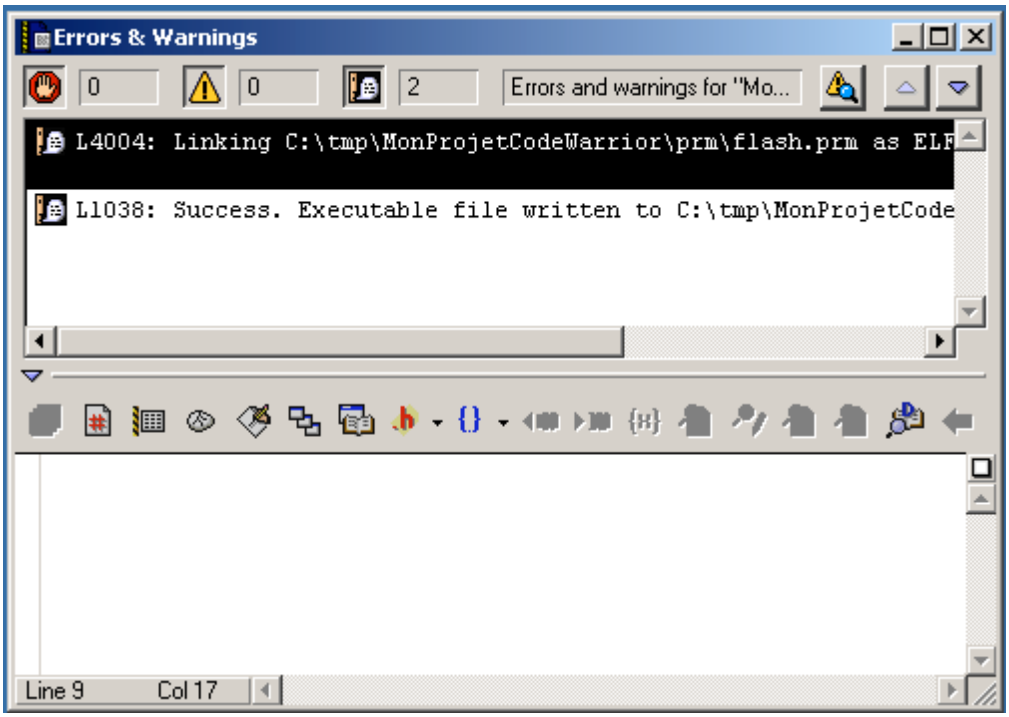
A.5 Comprendre les messages et corriger les erreurs

Ouvrir le fichier main.c. Modifier dans le source du programme le texte affiché par l'appel de fonction LCD_sendstring("STAR12 - FLASH"); afin qu'il affiche votre nom à la place de STAR12 - FLASH.
Régénérer l'exécutable : *Project -> Make* ou [F7] ou .


Errors & Warnings
Errors and warnings for "Mo...
Warning : C1801: Implicit parameter-declaration
main.c line 9
L4004: Linking C:\tmp\MonProjetCodeWarrior\prm\flash.prm a
Link Error : L1822: Symbol STAR12_initg in file c:\tmp\m
#pragma MESSAGE DISABLE C1420 // C1420 = "Result o
void main(void){
STAR12_initg();
puts ("Hello World ! \n\r");
LCD_sendstring ("STAR12 - FLASH");
LCD_gotoxy(0,1);
Line 9 Col 1

La fenêtre des messages apparaît. Elle contient des informations des avertissements (warning) et des erreurs (Error).
Ces messages sont provoqués par le compilateur (numéro commençant par un C) ou par l'éditeur de liens (numéro commençant par un L comme linker).

Au moins deux messages de l'éditeur de liens doivent apparaître. Lorsque le dernier indique que l'exécutable a été correctement créé, vous pouvez poursuivre en lançant le débogueur. Sinon, corrigez vos erreurs. Si des avertissements sont présents, vous devez les lire attentivement pour vérifier qu'ils ne correspondent pas à une erreur.


Errors & Warnings
Errors and warnings for "Mo...
L4004: Linking C:\tmp\MonProjetCodeWarrior\prm\flash.prm as ELF
L1038: Success. Executable file written to C:\tmp\MonProjetCode
Line 9 Col 17

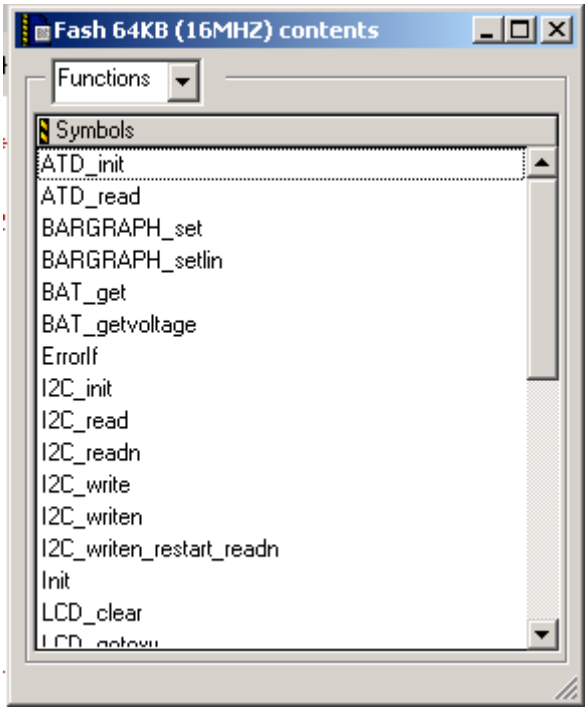
Note : Il est possible de désactiver ou de réactiver des messages dans le source avec la directive pragma MESSAGE,
par exemple
#pragma MESSAGE DISABLE C4000 // Desactive le message *Condition always is true*
#pragma MESSAGE WARNING C4000 // Reactive le message *Condition always is true*

A.6 Trouver la définition d'une variable ou d'une fonction

Ouvrez la liste des identificateurs (Window->Browser Contents) ou .
Vous y trouverez notamment toutes les fonctions définies dans le projet pour cette cible. Chaque élément est cliquable et ouvre ainsi le fichier correspond au bon endroit dans l'éditeur.

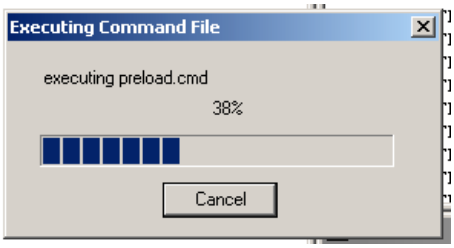
Dans l'éditeur, un clic droit sur un identificateur ouvre un menu contextuel permettant de voir sa définition.

A partir de la liste des identificateurs ci-contre, trouvez la définition de la fonction ErrorIf. Dans la fonction ErrorIf, avec le menu contextuel, trouvez la définition de la fonction LCD_init. De même, dans la fonction LCD_init, ouvrez la fonction TIMER_wait. Revenez en arrière avec le bouton « Back ».
Fermer toutes les fenêtres sauf celle du projet et la barre de menu CodeWarrior.



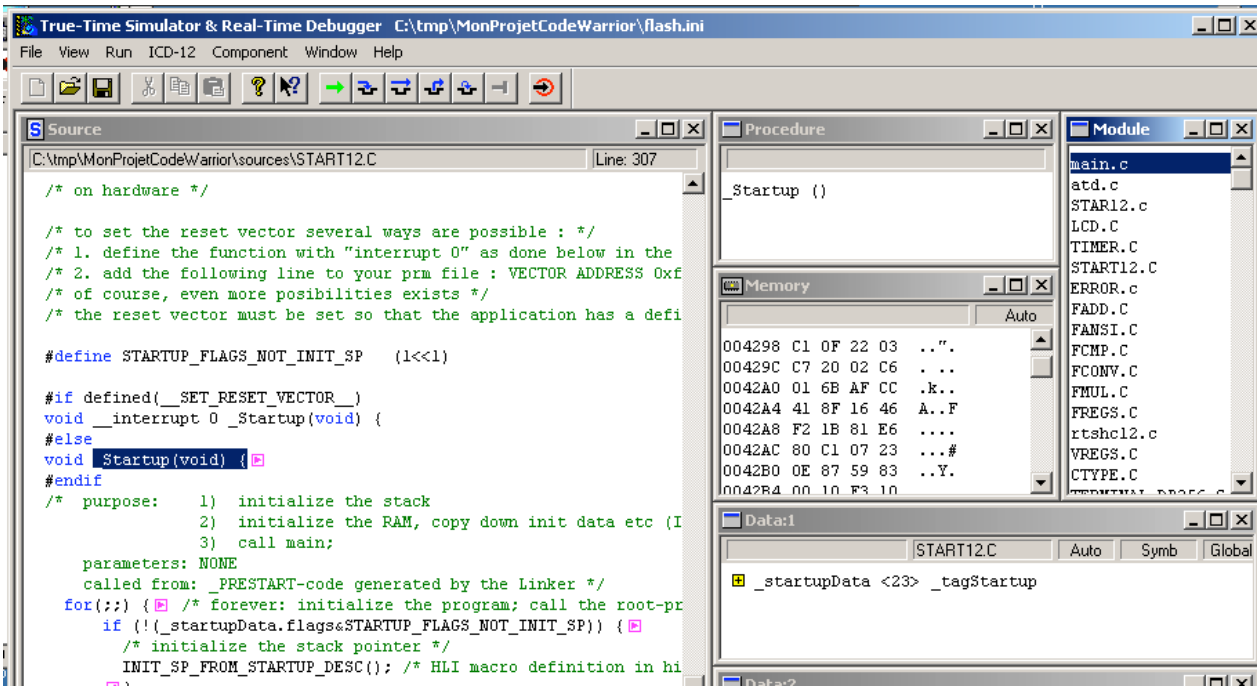
A.7 Lancer le débogueur HIWAVE et charger le programme en mémoire

A partir de la fenêtre du projet *: Project -> Debug* ou [F5] ou .
Si la compilation est nécessaire, elle s'effectuera. Le débogueur HIWAVE se lance, se connecte au boîtier BDM (s'il est bien connecté à la carte — qui doit être alimentée — et au port parallèle du PC), puis au micro lui-même et enfin charge le programme en Flash.



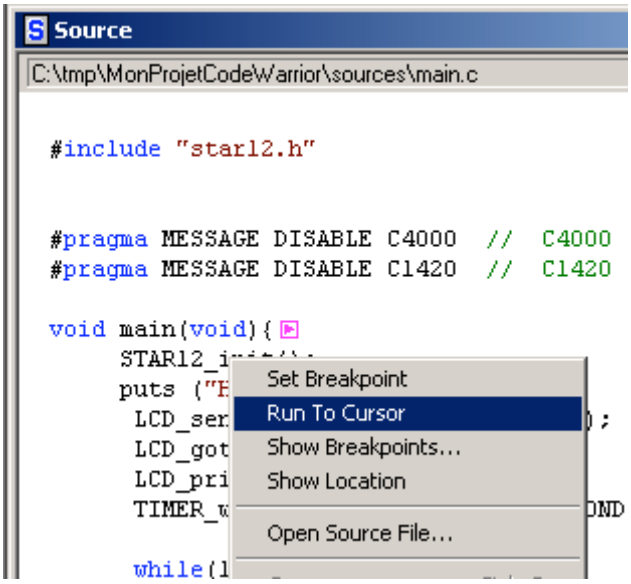
Cette étape est délicate et il ne faut pas intervenir (ni sur le PC ni sur la carte) pendant que le chargement s'effectue, même si le chargement se passe manifestement mal. En cas de problème, voir la rubrique dépannage à la fin de ce document. Si rien ne se passe, quand vous cliquez sur n'insister pas, consulter la rubrique dépannage.

A.8 Sauter le module de démarrage *startup* pour arriver dans la fonction principale *main()*



Le programme n'est pas encore lancé. Avant de le lancer, dans la fenêtre module, double cliquez sur le fichier main.c pour le faire apparaître dans la fenêtre de l'éditeur.

Ensuite, dans la fenêtre de source, par un clic droit sur le mot *main*, ouvrez le menu contextuel et sélectionnez "Run To Cursor". Vous éviterez ainsi de vous perdre dans le module de démarrage *startup*.



A.9 Contrôler l'exécution du programme

Vous pouvez avancer en pas à pas ou lancer/arrêter l'exécution du programme ou redémarrer la carte STAR12 avec les icônes de la barre d'outils:

	[F10]	Step Over	Avancer d'une ligne, sans rentrer dans les fonctions.
	[F11]	Single Step	Avancer d'une instruction en rentrant à l'intérieur des fonctions.
	[MAJ+F11]	Step Out	Finir la fonction en cours ... si la fonction a une fin !
	[CTRL+F11]	Assembly Step	Exécuter une instruction assembleur.
	[F5]	Start/Continue	Lancer l'exécution.
	[F6]	Halt	Arrêter le programme en cours STAR12.
	[CTRL+R]	Reset Target	Redémarrer le programme sur la STAR12.

Le coin inférieur droit vous indique dans quel état se trouve la STAR12.
Vous devez savoir à tout moment si votre programme est en cours d'exécution ou arrêté.

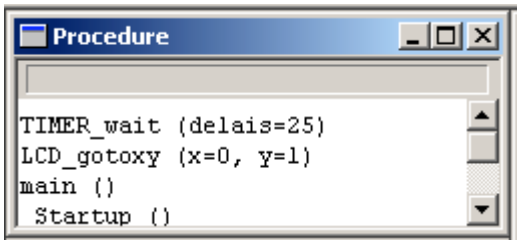
		Le programme s'exécute. Il peut être arrêté par le bouton ou par un point d'arrêt.
		Le programme a été arrêté. Vous pouvez placer un point d'arrêt, observer ou modifier la mémoire ou des variables, progresser en pas à pas ( ). Le programme peut être relancé par .
		Le programme est planté. Il a essayé d'interpréter une zone de mémoire qui ne contient pas d'instruction. Il faut redémarrer le programme

Lancez votre programme. Vérifiez qu'il affiche votre nom sur la première ligne de l'écran LCD. Tournez le potentiomètre et appuyez sur les boutons. Que se passe t'il ?

A.10 Imbrication des appels de fonction

Cette fenêtre présente l'imbrication des appels de fonctions et les argument d'appel transmis.

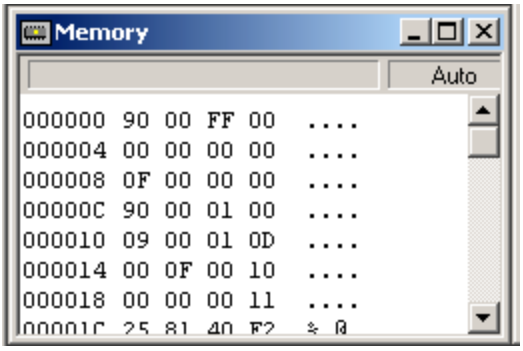
Ici, la fonction TIMER_wait a été appelée avec l'argument 25 par la fonction LCD_gotoxy, qui a été appelé avec les arguments 0 et 1 par main, qui a été appelé par la fonction de démarrage Startup.



Arrêtez le programme. Cliquez sur une fonction dans la fenêtre *Procedure* pour voir son code.

A.11 Manipuler la mémoire

Cette fenêtre représente l'état de la mémoire. La colonne de gauche représente les adresses. Les autres colonnes représentent des données. Par défaut, le contenu de la mémoire est présenté en hexadécimale, mais un menu contextuel (bouton droit) permet de choisir d'autres formats d'affichage.



Changez la valeur de l'octet situé à l'adresse 0000. Que se passe t-il sur la carte lorsqu'on met la valeur 00 ? FF ? 55 ? Faites de même avec l'adresse 250. Quels sont les bits modifiables ?

Il faut double cliquer sur une donnée pour la sélectionner en édition. Il faut ensuite valider la modification avec la touche entrée.

A.12 Placer un point d'arrêt

Dans la fenêtre module, sélectionnez STAR12.C. Avec le menu contextuel (bouton droit), placez un point d'arrêt (Set Breakpoint) sur la fonction *BARGRAPH_set.* Le point d'arrêt se matérialise par une flèche.

```
    DDRJ = 0x02; // PORT J1 connected to L
    DDRS = 0x00;
}

voi BARGRAPH_set (UINT16 value){
    UINT8 bit8;
    UINT8 bit9;
    bit8 = (value & (1<<8)) == (1<<8);
```

Lancez le programme ( ). Il doit s'arrêter sur ce point d'arrêt.

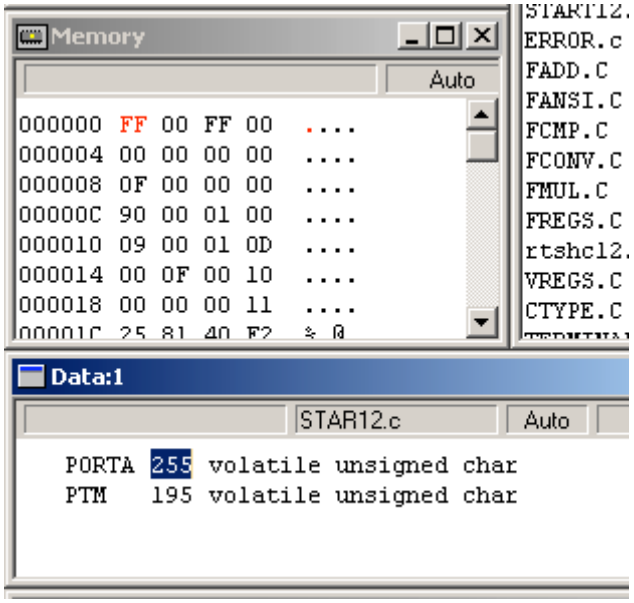
```
    DDRS = 0x00;
}

voi BARGRAPH_set (UINT16 value){
    UINT8 bit8;
    UINT8 bit9;
    bit8 = (value & (1<<8)) == (1<<8);
```

A.13 Observer et modifier des variables

Dans la fenêtre des variables globales (Data:1) repérez les variables PORTA et PTM.

Essayez de changer la valeur de PORTA et PTM. Que se passe t-il dans la fenêtre mémoire aux adresses 0000 et 0250 ? Sur la carte ?



B Manipulations

B.1 A vos marques … prêt … affichez

Maintenant, fermez le débuggeur HIWAVE, fermez l'environnement CodeWarrior et cachez vos notes. Vous allez, le plus rapidement possible, créer et faire tourner sur la carte STAR12 un projet nommé HC12BIDON dans le répertoire c:\tmp qui affiche "CACHAN THE BEST " sur la première ligne de l'écran LCD.

- Si vous mettez plus de trois minutes, vous êtres trop long. Recommencez avec un nouveau projet HC12_BIDON2 qui affffiche « 2eme essai» sur l'écran LCD.
- Si vous mettez moins de 1 minute 45, vous êtes bon. Félicitation ! C'est l'heure de la pause.
- Le record officiel homologué est de 1 minute et 2 secondes.
- L'auteur du sujet un BOUNTY au premier qui descendra sous la minute !

Pause …

B.2 Lire l'état des boutons et imposer l'état des LED

Développer une application qui reproduit en permanence sur les quatre LED du bas l'état des quatre boutons poussoir. Dès qu'un bouton est enfoncé, la LED correspondante doit être allumée. Vous pouvez utiliser les macros PORTA et PTH définies dans le fichier 6812dp256.h.

Les LED sont connectés au port A et les boutons au port H. Vous pouvez remplacer le code présent dans la boucle infinie du programme stationnery.

B.3 Pointeurs et adresses

Développer la même application, mais cette fois-ci sans utiliser les macros PORTA et PTH définies dans le fichier 6812dp256.h. Vous devez définir vos propres pointeurs.

Le registre PORTA se trouve à l'adresse 0000H. Le registre PTH se trouve à l'adresse 0260H. Un fiche décrit l'utilisation des pointeurs pour accéder à une adresse.

B.4 Masque de configuration 0-1-X

Coder une application qui répond au cahier des charges suivant :
Allumer la seule LED N°3 lorsque le bouton vert est enfoncé. Les autres LED doivent rester éteintes. La LED n°3 doit s'éteindre dès qu'on relâche le bouton vert. Les autres boutons ne doivent pas avoir d'effet, sauf le bouton blanc qui ne doit pas être enfoncé sinon la LED s'éteint.
Vous pouvez utiliser les macros PORTA et PTH définies dans le fichier 6812dp256.h.

Quelle configuration du port H permet d'allumer la LED 3 : (0,1 ou x pour chaque bit) ?

	7	6	5	4	3	2	1	0
PTH =								

B.5 L'application à 23 caractères

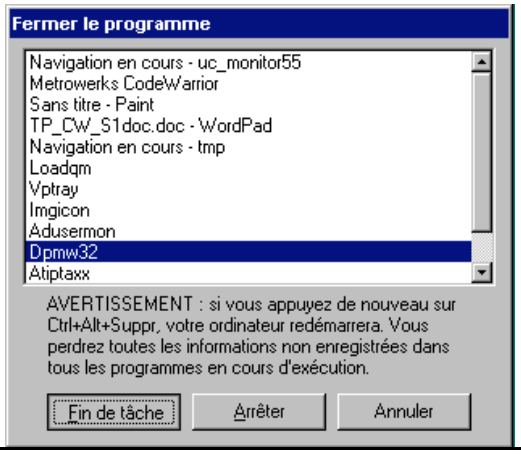
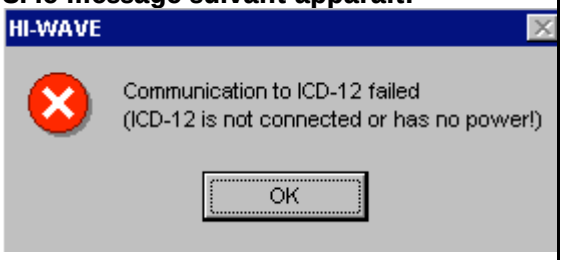
Coder la même application mais en respectant les contraintes suivantes :
En dehors de la boucle infinie : Tout est permis.
Dans la boucle infinie :

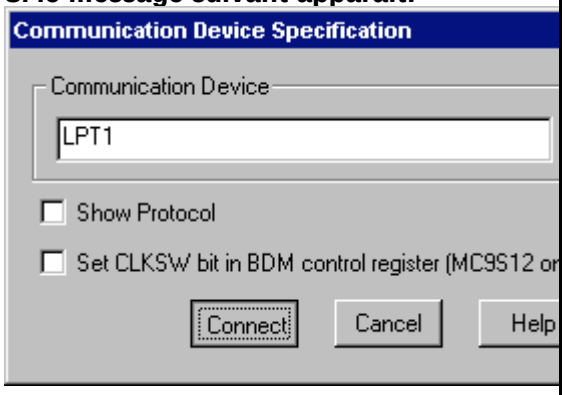
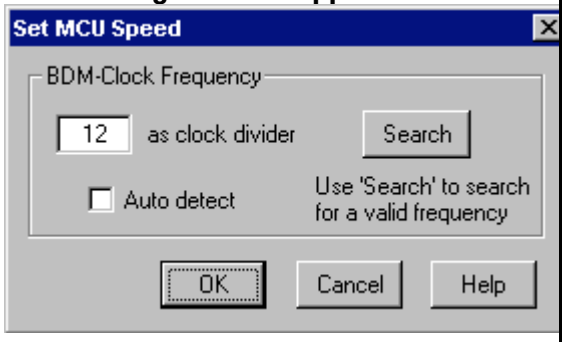
- Vous ne devez pas utiliser d'appel de fonction.
- Vous ne devez pas utiliser de macro (#define).
- Vous devez utiliser le moins possible de caractères à l'intérieur de la boucle infinie.
- Les espaces et retour à la ligne ne comptent pas.

Le record actuel est de 23 caractères pour le corps de la boucle, accolades comprises.

— FIN —

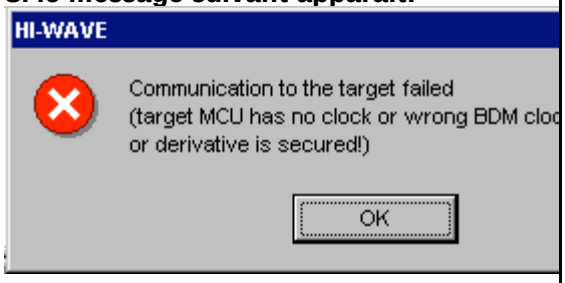
C Dépannage

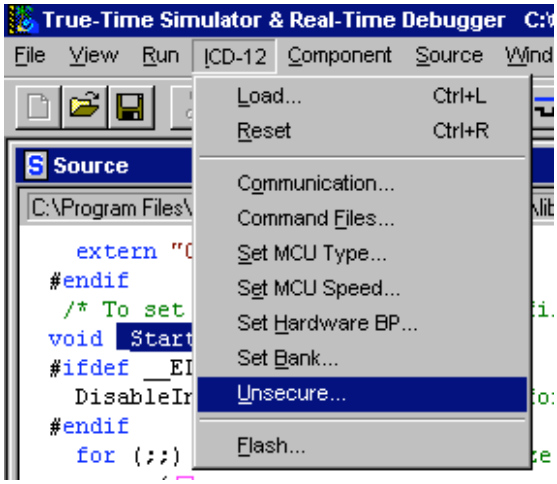
FUSIBLE	**La LED de l'alimentation sur la carte STAR12 est éteinte et rien ne s'allume quand on met la carte sous tension.**	Vérifiez le fusible. S'il a fondu, demandez en un autre à votre enseignant ou aux techniciens.
DPMW32	**Si le débogueur ne se lance pas quand vous cliquez sur** ... 	Sous Windows 95/98, il arrive fréquemment que le débogueur HIWAVE ne se lance pas. C'est un bug. N'insistez pas. Faites [CTRL-ALT-SUPPR] une seule fois pour faire apparaître la fenêtre des tâches. Sélectionner la tâche nommée *Dpmw32*. Forcer la tâche à s'arrêter en cliquant sur *[Fin de tâche]*. Le débogueur doit maintenant se lancer. Si vous avez insisté en essayant plusieurs fois de lancer le débogueur , vous vous retrouvez maintenant avec plusieurs débogueurs ouverts, parfois une bonne dizaine. Ils tentent tous d'accéder à la carte simultanément, ce qui, bien sûr, n'est pas possible. Fermer les tous et n'en relancez qu'un seul.
NO CONNECT OR NO POWER	**Si le message suivant apparaît.** 	Vérifier la connexion du connecteur ICD12 (BDM Multilink) vers le PC et vers la carte ainsi que l'alimentation de la carte. Valider le message par le bouton [OK].

LPT1	**Si le message suivant apparaît.** 	Se connecter à nouveau en validant le bouton [Connect].
CLOCK DIVIDER	**Si le message suivant apparaît.** 	Cliquer sur le bouton [Search]. HIWAVE doit vous proposer une valeur de diviseur d'horloge raisonnable. Classiquement, les bonnes valeurs sont 12 ou 13 avec un oscillateur à 16MHz et 2 ou 3 avec un oscillateur à 50MHz.

WRONG CLOCK OR SECURED

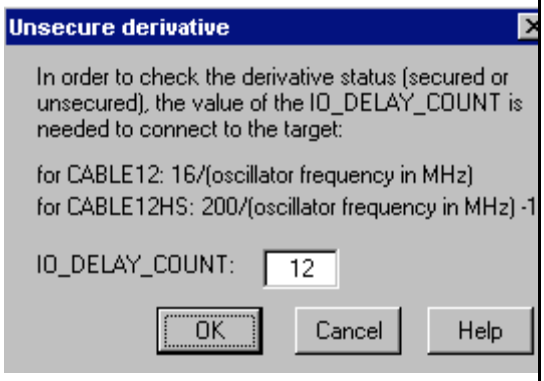
Si le message suivant apparaît.





Le problème est peut-être sérieux. Appelez un enseignant.

En premier lieu, si votre composant est alimenté sur batterie, vérifiez le niveau de charge de la batterie.
Ensuite vérifier l'état du câble en nappe 6 broches du BDM.
Enfin, tenter de "désécuriser" le composant. Dans HIWAVE, menu ICD-12, item unsecure...


Unsecure derivative
In order to check the derivative status (secured or unsecured), the value of the IO_DELAY_COUNT is needed to connect to the target:
for CABLE12: 16/(oscillator frequency in MHz)
for CABLE12HS: 200/(oscillator frequency in MHz) -1
IO_DELAY_COUNT: 12
OK
Cancel
Help

En cas d'échec, me signaler le problème.
Vous pouvez essayer de changer
- d'abord le câble 6 points du BDM,
- puis le boîtier BDM lui même
- enfin la carte star12.

Si du matériel est effectivement défectueux, merci de me le signaler.

:
: